An Approach to Dynamic Provisioning of Social and Computational Services

Luiz Olavo Bonino da Silva Santos, Vikram Sorathia,
Luís Ferreira Pires and Marten van Sinderen

University of Twente

Enschede, 7500 AE

The Netherlands

Email: l.o.bonino, v.s.sorathia, l.ferreirapires, m.j.vansinderen@ewi.utwente.nl

Abstract—Service-Oriented Computing (SOC) builds upon the intuitive notion of service already known and used in our society for a long time. SOC-related approaches are based on computer-executable functional units that often represent automation of services that exist at the social level, i.e., services at the level of human or organizational interactions. With the increasing adoption of SOC, more complex scenarios mixing computational and social services are emerging, raising the need to better understand the relationship between social and computational services and to specify and model them accordingly. In this paper we present our ontology-based approach to dynamic service provisioning. Our approach aims at improving the current state of the art by allowing an explicit distinction between social and computation services and dynamic service provisioning support for these two types of services. We illustrate the applicability of our approach with a use case scenario in the health care domain.

I. INTRODUCTION

Service-Oriented Computing (SOC) [1] and Service-Oriented Architecture (SOA) [2] treat services as location-transparent, platform-independent and self-contained functional units that can be accessed through a set of standard protocols and technologies. SOC and SOA efforts mainly focus on technological issues, such as service discovery and composition algorithms, semantics for computer inference, computer-readable service descriptions, among others. However, apart from some inherently computer-related services such as data transformation services, bandwidth provisioning services, etc. what people normally call a *service* in the scope of SOC is actually an (full or partial) automation of services that exist at the social level. For instance, an online travel booking service is an automation of the travel booking services usually provided by travel agencies, airlines, hotels, car rentals and other travel-related organizations in the real world.

With the increase of SOC adoption, more complex scenarios with a large number of available (social and computational) services emerge, so that a number of issues have to be sorted out, such as: (i) the identification and modeling of services at the computational or at the social level; (ii) the degree of automation of social services by computational services; (iii) the modeling of domains with services at both levels mixed; and (iv) the support of these domains by a

software platform.

In this paper we present our approach to dynamic service provisioning that aims at tackling the aforementioned issues. This approach is supported by an upper-level ontology, a domain modeling language derived from this upper-level ontology and a supporting software platform. The upper-level ontology and its derived domain modeling language define concept and relation primitives that allow domain specialists to specify service-oriented domain ontologies. The domain ontologies are then fed into a software platform to support users in the dynamic service provisioning activities.

This paper is further structured as follows. Section II discusses the concept of service, and the distinction and relations between social and computational services. Section III gives an overview of our approach to dynamic service provisioning. Section IV presents a use case scenario in the Health Care domain. Section V presents and discusses the modeling of the use case scenario using our approach and the software platform we have been developing. Section VII gives conclusions and final remarks.

II. SOCIAL AND COMPUTATIONAL SERVICES

From the last few years we have been observing an increasing adoption of the service paradigm in Computer Science with the consequent growth on the number of research and industry efforts to develop technologies and techniques that support its use. Examples of research efforts in the field of SOC are proposals for reference architectural pattern for SOC implementations and proposals for service ontologies, and [6], [7], aiming at providing a shared conceptualization for services. A commonality among these efforts is that they consider services as building blocks to realize business processes. The relationship between services and business processes are perceived in a limited distinction between social and computational levels. In these efforts, the social level is often represented by the business processes and a computational level represented by the (Web) services. The business process (or the part of it that is to be automated) is then mapped into a computer-executable process. In our work, we claim that this distinction should be further explored so that the dichotomy between social

services and computational services, and the relationships between services at each level should be made explicit.

Our claim is motivated by the assumption that from now on we will be involved in an increasing number of scenarios with a complex mixture of interrelated computational and social services. With the complexity increase in such service-oriented scenarios, an explicit and clear distinction between social and computational services becomes necessary. Clarifying this distinction should enable software provisioning platforms to support not only the provisioning of computational services but also (in some degree) social services, and combinations of computational and social services. When business processes are mapped onto computer-executable processes, the relationship between social and computational services are hard-coded. In this mapping, the computational service becomes the business process and any change on one implies a change on the other. Contrarily, if we can manage the distinction between social and computational services and how they related to each other we would be able to establish different automation levels, supporting dynamic reconfiguration, evolution, composition, etc.

Additionally, when faced with a small number of services (either social or computational), service clients are capable to carry out service provisioning steps manually, namely, discovery, selection, negotiation, contract and invocation. However, given a large amount of services it becomes unfeasible to carry out these steps manually, raising also the need for a software platform that supports these service provisioning steps.

In order to tackle multi-level (social and computational levels) service identification, specification and modeling and dynamic service provisioning, we start with a brief analysis of the concept of service. As presented in [8], the term service has been frequently used in Economy to represent immaterial and intangible products, while the term goods has been used to represent its material and tangible counterpart. To avoid this kind of misunderstanding, in [8] the authors present an historical overview of the definitions of services and goods, and after discussing their relevant characteristics provide a taxonomy that differentiates these two concepts, being goods either material and tangible (e.g., a computer, a chair, etc.) or immaterial and intangible (e.g., a software program, a musical composition, etc.). In [8], a service has one fundamental characteristic, namely the mandatory existence of relationships between producers and consumers. The authors claim that the idea of one entity (the service producer as it is called in this paper) acting for the benefit of another entity (the service consumer) is inherent to services. Moreover, contrarily to goods, services are not entities that exist independently of their producers and consumers and are defined as “some (material or immaterial) change in the condition of one economic unit produced by the activity of another unit”. Although we agree with the arguments given in [8] we believe that their definition of service is incomplete

and does not cover all the aspects defended by the authors, especially the explicit and mandatory relationship between the service producer and the service consumer.

In research efforts such as [9] and [10], the authors aim at providing ontological foundations for services and service science, respectively. The work in [10] provides an initial step towards a rigorous and principled understanding of *Services Science* based on principles of ontological analysis. In [10], multi-level services are also considered, which are services at the social level having computational services (or e-services, as called in the paper) ultimately providing social benefits. In [10] the definition of service is given as: “A service is present at a time t and location l iff, at time t , an agent is explicitly committed to guarantee the execution of some type of action at location l , on the occurrence of a certain triggering event, in the interest of another agent and upon prior agreement, in a certain way.”. We consider this ontological analysis relevant to the area of SOC as it provides a clear understanding of the service concept and the intrinsic characteristics of a service. This definition also enables the formalization of the service concept to be used in computer-readable artifacts.

In our work, we aim at distinguishing and clarifying how social and computational services relate to each other. As defined in [10], we consider that a service encompasses a set of commitments that determine that a service provider performs a task for the benefit of a service client under certain conditions. We have identified two main types of relations between social and computational services, namely, (i) a computational service (fully or partial) automates a social service, or (ii) a computational service supports one or more of the service provisioning steps of a social service.

The automation relation between a computational service and a social service derives from the actual automation of the task related to the social service, done by the task related to the computational service. For instance, an online hotel booking service performs a computational service task that interacts with a set of hotel-related databases and returns availability, price and booking confirmation based on given dates, location and room parameters. This computational service automates the social hotel booking service that provides equivalent results by performing its task at the social level.

For some services it may be impossible to automate their core functionality exclusively with computational services. For instance, the air transportation service that moves people or goods from one location to another cannot (yet) be automated by computational services. However, in this case some steps of the service provisioning still can be supported by a software platform. For example, although the core functionality of the transportation service cannot be automated, some of its provisioning steps, such as discovery (e.g., through flight search services), negotiation (e.g., through online flight booking services) and activation (e.g., through online check-

Figure 1. Framework for provisioning support of multi-level services

in services), are often supported by computational services.

III. A GOAL-BASED SERVICE PROVISIONING FRAMEWORK

In this section we briefly present our service provisioning framework in order to introduce the conceptual and technological support we have devised for our service provisioning approach. To support dynamic provisioning of social and computational services we proposed the Goal-Based Service Framework (GSF) [11]. GSF is based on goal modeling and assumes that the involved stakeholders (service clients, service providers, context providers) share the same conceptual models, i.e., the same set of domain ontologies. Figure 1 depicts the main elements of our framework. Each element is described as follows:

- *Goal-Based Service Ontology (GSO)*. This ontology defines domain-independent concepts such as service, social service, computational service, service client, service provider, goal, task and their relations, among others. GSO extends the Unified Foundational Ontology (UFO) [12] by adding the service-related concepts and relations.
- *Goal-Based Service Metamodel (GSM)*. Generated from Goal-Based Service Ontology, this metamodel represents the concepts defined in GSO and defines the language used by domain specialists to create domain ontologies.
- *Domain Ontologies*. GSF can be used in different application domains such as health care, ambient intelligence, etc. For each of these domains, domain specialists define a domain ontology, namely the concepts and relations relevant to the domain, goals that users can have, valid tasks in the application, etc. Domain specialists are ontology engineers together with experts of the domain, e.g., physicians in the case of the health

care domain. GSM defines a modeling language that enables domain specialists to define domain ontologies, allowing knowledge about particular domains to be shared.

- *Context-Aware Service platform*. The context-aware service platform supports the interaction between service providers and service clients. From the service provider's perspective, the platform supports the publication of service descriptions. From the service client's perspective, the platform provides mechanisms for service discovery, composition, invocation and monitoring, among others. Moreover, the context-aware components of our supporting platform gather users' contextual information that is used (i) to filter the discovery and selection of candidate services and, (ii) as input data for the selected services. Context information gathering reduces the need of direct user input and, thus, reduces also the need of users' interaction allowing the platform to be more autonomous.

A standard deployment of GSF consists in GSO, GSM and the Context-Aware Service Platform (CASP). Domain ontologies can be added to the CASP by domain specialists. Whenever domain ontologies are available in the CASP, service providers can semantically annotate their service descriptions based on the concepts defined in these domain ontologies. Service descriptions are added to the CASP by the service providers.

IV. A HEALTH CARE USE CASE SCENARIO

We motivate and illustrate the applicability of our approach with a usage scenario in the health care domain. This domain is suitable to exemplify the distinction between social and computational services and their two types of relations discussed in Section II, since many core services in this domain require human intervention and cannot be fully automated. The scenario presented in this paper has been based on common cases of frequent traveler professionals who may require medical assistance while in a business trip.

In our scenario, Lucia is a professional whose job requires frequent business trips. In one of these trips, she has a health-related event and needs to consult with a doctor. However, since she is not near her house and, consequently, far away from her general practitioner, she needs to find a nearby doctor that complies with her health insurance's conditions. Figure 2 shows the stakeholders and resources involved in this health care situation. The main transaction is between Lucia and Dr. Smith, the discovered general practitioner. Other stakeholders and resources have also been identified and are depicted in Figure 2.

The four distinct aspects of our health care scenario are:

- *Health service provider configuration*. A provider configuration is a collection of roles and resources that are collectively responsible for a health care service provi-

Figure 2. Health care scenario

sion. The configuration is the result of an organization activity, appropriately governed and managed.

- *Partners and enablers.* These are entities (e.g., organizations) that are not directly involved in the service provision but are essential for the service operation. Pharmacies and medical laboratories are examples of such entities.
- *Service client.* In this scenario a service client is a nomadic professional who may seek medical assistance while on the move. Although in our upper-level ontology we distinguish between the service client (the one who pays for the service) and the service beneficiary (the one who benefits from the service), for simplicity we consider here that the service client is also the service beneficiary.
- *Business environment.* The business environment consists of other entities that are relevant not from an operational point of view but from the strategy and policy viewpoints. Examples of such entities are (i) the registry where service offerings are published, (ii) the governmental health authority that is responsible for setting various policies, rules, guidelines and procedures that must be observed, and (iii) competitors, whose activities should be closely observed in order to determine next steps of the marketing strategy and (possibly) adjustments on service offerings.

In our scenario we consider the following types of service providers:

- *Remote Patient Monitoring Services Providers* allow the current situation of a patient to be monitored by sensors. These sensors can monitor health parameters like, for example, blood pressure, glucose and ECG, and are typically deployed as wearable instruments connected in personal area network (PAN). The readings of these instruments are transmitted to the associated health

care providers, so that whenever the values measured by these instruments falls outside a pre-defined range triggering conditions for responsive activities can be generated.

- *General Practitioners (GP)* are generally the first points of interaction in the health system. They evaluate the health condition by preliminary checkups and may recommend detailed medical tests, prescribe medicines, or refer to the expert service providers.
- *Medical Experts* can be involved depending on the preliminary observations by GP. Medical experts are specialists like cardiologists, neurologists, pediatricians, etc. who are consulted for specific types of health problems.
- *Pharmacies* provide drugs prescribed by general practitioners or experts as well as freely-available drugs.
- *Medical Laboratories* are important partners in medical services. GPs or specialists request tests that are carried out by medical laboratories. The test results are used by the physicians to support their diagnosis. Tests may include red cell counting, cholesterol levels, X-rays, MRIs, etc.
- *Medical Insurance Services Providers* provide medical (health) insurance as a financial service designed to cover the expenses of a patient (health service client) when accessing health services. The financial arrangements among the insurance company, the client and various health service providers allow health services to be consumed possibly without paying any extra money for individual transactions. In several countries health insurance is compulsory, so that every inhabitant of these countries can be considered as a client of a medical insurance service.
- *Other health service providers* such as dentists, physio-therapist and dietitians. These health service providers offer services that can help improve the overall health of a patient, and are often available in a pool of providers (e.g., a directory of certified health service providers). Patients are generally free to choose from the available pool of service providers.

Considering Lucia as the patient, accessing health services can be quite challenging in a mobile situation. Once a health-related event occurs, the selection of service providers may be difficult, especially in an unknown geographical region. A directory look-up may result in a list of service providers, so that Lucia has to manually evaluate and select a suitable one. For example, her international insurance package may not be recognized by the nearest local pharmacist. Furthermore, her health records and history may not be accessible by the GP she is consulting. Therefore, selecting those providers that work seamlessly and accept her insurance package can be quite challenging, as this information cannot be found in any regular directory listing service but has to be inferred

Figure 3. Social and computational services

somehow.

From the service provider point of view, mobile professionals can be an important part of the potential clients. Service providers can identify possible interoperability issues that may improve their chances of being selected. They may have to work out arrangements with other service providers within their area and also with the providers abroad so that they can seamlessly provide the service. In their service offerings, they can indicate the standards and protocols they support and the services with which they interoperate.

V. HEALTH-CARE DYNAMIC SERVICE PROVISIONING

Inspired by the use case scenario introduced in Section IV, we present here our approach for modeling the health care domain and supporting dynamic service provisioning. In our framework, domain specialists are expected to model the domains and provide the resulting domain ontologies to the framework’s service supporting platform. To model a domain, a domain specialist should use the language derived from the framework’s upper level ontology, namely, the Goal-Based Service Ontology (GSO) [13].

Figure 3 shows a GSO excerpt that depicts the relation between computational and social services. In GSO, services (commit to) perform tasks. A *Service Task Type* is a subcategory of *Action Type*, which can be instantiated by a *Action* (an individual) creating a *Situation*. This *Situation* (state of affairs) is the outcome of a service execution and can be related to what is commonly referred in SOC as service effects. The service task also requires some inputs and creates outputs. In our use case scenario, medical consultation, medicine delivery and medical test performance are modeled as *Social Services*. The medicine delivery social service requires a medical prescription as input type and its execution creates a situation where the service client has access to the prescribed medicine. Similarly, the medical test performance social service requires the input of a medical request, creates some information as test's result, and its execution creates a situation where the doctor has access to the results.

Figure 4 shows another GSO excerpt, which depicts the relations between *Service Type*, *Service Client Type* and *Service Provider Type*. Services types can be specified in

Figure 4. Service provisioning events

a given domain by associating them with service providers types that can offer certain kinds of services, and service clients types that can request these kinds of services. Service clients and service providers are related by the agreements concerning service provisioning. This relation can be generated whenever a service client finds a service whose related task creates a situation that satisfies the service client's goal. The *Service Provision Event Type* represents types of events that can contribute to or enable service provision, such as *Service Negotiation Type*, *Service Discovery Type* and *Service Activation Type* (for the sake of space, not all events are depicted in Figure 3). When a service client discovers and selects a service, a negotiation takes place to determine the conditions and constraints for the service provisioning. A successful negotiation creates a service agreement type, which is a social relator that binds the service client and service provider, and can be potentially composed out of a set of commitments and claims, e.g., the commitment of providing the service under certain conditions and for an specified cost. This social relator (the *Service Agreement Type*) can be described in a contract (omitted in Figure 3) which is a normative description [14].

In our use case scenario, a patient is a *Service Client Type*, while health insurance companies, pharmacies, general practitioners, medical experts and laboratories are *Service Provider Types*. When closing the deal for a health insurance, the agreements between the health insurance company and the patient are realized in a health insurance contract, which is a *Service Agreement Type* in our health care domain.

Figure 5 depicts how services are described. In GSO, we consider that different parts of a service have different descriptions. The *Service Profile* provides an overview of the service for advertisement purposes. It describes what the service does (in a human readable form), its requirements and conditions. Also, service-level agreement parameters can be included and used in the service negotiation. The *Service Model* describes the Service Task and provides information about the activities involved in the *Service Task* execution. The *Service Model* is used to assess the service's behavior, i.e., what set of activities the service performs. The Service

Figure 5. Service description

Model can be used for service monitoring and orchestration (not discussed in this paper). Moreover, the Service Model can be described in different granularity levels allowing a more superficial or more in-depth view of the *Service Task*. The *Service Grounding* describes the *Service Interface*. The *Service Interface* defines information necessary to invoke the service, i.e., to trigger service execution. In the case of *Computational Services*, the *Service Interface* defines the technology-specific information necessary to invoke the service, namely, the communication protocol, parameters' types, URI, etc. GSO does not commit to any particular language to describe services, such as WSDL, OWL-S or SAWSDL. Any of these languages could be used to instantiate the concepts defined in GSO.

Various functional and non-functional parameters of services can vary depending on the domain. Hence, domain specialists can extend basic service concepts defined in GSO to define domain-specific vocabularies for describing services in their respective domains.

VI. USE CASE SUPPORT

A health care service transaction may span multiple service providers. A service client may select and access the service of any of the available providers. As depicted in Figure 6, once a health condition is established, the patient (service client) can decide to consult a GP. There can be many GPs in a geographical vicinity. The patient can select a practitioner by evaluating criteria like geographical proximity, availability of earliest appointment or reputation. During the consultation, the GP may request additional tests. From the pool of medical laboratories, the patient may again need to select a provider based on various criteria. The tests' results are evaluated by the GP, who prescribes the treatment and possibly some medication drugs. Once again, the patient needs to choose a provider from a pool of pharmacists. As the patient is receiving the health service, the financial aspects are covered by her insurance company, which she has selected much earlier from an insurance service providers pool.

Hence, in this typical health care scenario, the actual service provision may include several providers. Each provider can be selected independently, but the services they provide may fulfill specific pre-conditions. For instance, laboratories typically require a test request from a GP or Medical Expert

Figure 6. Service provider selection in a typical health care service consumption scenario

stating the test specifications. Here, the prescription can be seen as outcome of a service transaction carried out by the GP or Medical Expert. Similarly, pharmacies often require a drug prescription before they deliver the requested drug.

A. Domain ontology in health care

A health care domain ontology can be defined by specializing and instantiating concepts of GSO. For this purpose, GSO is imported as a reference ontology. With the concepts of *Service Provider*, *Service Client* and *Service Task* imported from the GSO, the health domain expert can model the *Health Service Provider*, the *Health Service Client* and the *Health Service Task*, respectively. Using the GSO's *Service Description* concept and its sub-concepts, health domain experts can define additional details related to the *Service Profile* and *Service Grounding*, by encoding the terms, conditions and methods necessary to perform health care-related tasks. For instance, a medical consultation service may require as inputs information about patient's medical history, number of medical policy, etc.

B. Configuration

Domain ontologies defined by the domain specialists are utilized as reference ontologies by the Context-Aware Service Platform (CASP). The health ontology and other domain ontologies defined in similar manner are stored by CASP in an ontology repository so that they can be referred to at runtime for semantic service provisioning. These ontologies are used to annotate service descriptions and to support service requests. In our prototype, we have used Sesame ²¹, which is an open source framework for storing, inferencing and querying ontologies. The Sesame Server allows one to build a repository that can be accessed through

²¹<http://www.openrdf.org>

Figure 7. Semantic query to determine service tasks for the given goal

the web-based OpenRDF Workbench or programmatically by using query languages such as SeRQL and SPARQL through the Storage And Inference Layer (SAIL) API.

C. Semantic service selection

Initially, the service client defines to the CASP that her main goal in the health care domain is to *stay healthy*. Assuming that the initial state of the service client is that she is healthy, the CASP can be configured to collect user's contextual information to identify any situation that may trigger the need for a health service. Remote patient monitoring technology can be used as an information source to collect user's health-related information to determine if the health service client is no longer healthy and, therefore, needs medical attention. Whenever the contextual information suggests the need for medical attention, the CASP triggers a query to identify all tasks that can be carried out to satisfy the given goal. All tasks that fully or partially satisfy the *stayHealthy* goal can be determined by the query depicted in Figure 7.

This query is performed on the health domain ontology and returns the service tasks *PerformMedicalDiagnosis*, *PrescribeDrugs* and *PerformMedicalConsultation*, which partially fulfill the *stayHealthy* goal. From the list of identified tasks, another semantic query can determine the tasks that can be performed first. The ordering is given by the task decomposition primitives of GSO, allowing the definition of structures of tasks and sub-tasks. The CASP determines that *PerformMedicalConsultation* is the first task to be considered and, for performing this task, appropriate service provider types should be identified. A semantic query can be performed to determine the service provider type *HealthServiceProvider* that can perform *ProvideMedicalConsultation*, as depicted in Figure 8.

There can be multiple instances of service providers offering the same service. The CASP can use the user context information to further refine the selection criteria. The service grounding of the *ProvideMedicalConsultation* service requires input parameters such as *Location* and *Sup-*

Figure 8. Semantic query to determine Service Provider Type for the given Service Task

portedPaymentMethod, which can be employed as selection criteria. Figure 9 indicates a SPARQL query that can be employed to identify the most suitable service provider for this service from the available pool. The query indicates that both GSO (marked as 1 in Figure 9) and Domain ontology (marked as 2 in Figure 9) are used. As individual domain ontologies are derived from the GSO's concepts, query of the various service parameters defined in GSO is allowed. The Service Grounding (marked as 3) defines the domain-independent property *hasSupportedPaymentMethod*, which can be filtered with the domain-specific parameter *MedicalPolicy101* (marked as 4) enabling consistent search strategies that are applicable in various service consumption scenarios.

Figure 9. Semantic query for Service Provider instance selection

The successful execution of this query indicates the selection of the first service provider in a health care consumption scenario depicted in Figure 6. Depending upon the outcome of *ProvideMedicalConsultation* task, the CASP can determine the next service task to be executed towards the fulfillment of the given goal. The same process can continue until the goal is fulfilled.

VII. CONCLUSION

This paper presents an approach for dynamic provisioning of multi-level services. In order to motivate our approach we discussed the distinction between social and computational services and how they are related. Our approach for dynamic

service provisioning is supported by an upper-level ontology from which a metamodel for a domain specification language is derived. Domain specialists can use this domain specification language to define domain ontologies in the scope of Service-Oriented Architecture. The domain specification language enables the modeling of social and computational services. A supporting service platform completes the framework offering a software infrastructure for dynamic service provisioning.

In this paper we also present a health care use case scenario, aiming at both motivating the need for a clear and precise distinction between social and computational services, and demonstrating how this distinction can be supported by software platform for dynamic service provisioning. The health care domain has been chosen for its widespread applicability. We have identified some challenging requirements for health care mobile communities and the issues faced by providers and clients due to mobility. We argued that appropriate architectural and modeling support can help solve these problems.

Some relevant parts of our upper-level ontology have been discussed and we show how this ontology can be used to model our use case. This framework is primarily targeted to scenarios where the service clients are end-users without deep technological knowledge or where the service clients require limited explicit interaction with the services. Social services seem to fit these characteristics, since we expect that clients of social services do not have knowledge on SOC-related technologies such as WSDL, SOAP and XML and furthermore do not want to be bothered with all the details of interacting with computational health services.

Our framework assumes the existence of domain ontologies in which domain and task concepts are defined. These ontologies are expected to be made available beforehand by domain specialists. This assumption makes the framework suitable for environments where the domain is clear and well known. Examples of suitable domains for our framework are Ambient Intelligence (AmI), health care and mobile pervasive applications, where users are not expected to interact too often with the computational devices.

Currently, the first version of GSO/GSM has been designed together with the Home Health Care domain specification. A prototype of the platform has also been implemented and tested with a limited number of services and ontology concepts. In our future work we intend to (i) define techniques, guidelines and tool support for client's goal specification and domain specification based on GSO; (ii) use model transformation techniques to automatically transform goals and tasks models into service requests; (iii) model additional domain ontologies to assess the appropriateness of GSO in multiple domains; (iv) test the platform with more complex domains and a larger number of services; (v) define evaluation criteria for the framework and; (v) evaluate the framework comprehensively according to the

defined criteria.

REFERENCES

- [1] M. N. Huhns and M. P. Singh, "Service-oriented computing: Key concepts and principles," *IEEE Internet Computing*, vol. 9, no. 1, pp. 75–81, January 2005.
- [2] T. Erl, *Service-Oriented Architecture: Concepts, Technology, and Design*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2005.
- [3] A. Arsanjani *et al.*, "Design an soa solution using a reference architecture," March 2007. [Online]. Available: <http://www-128.ibm.com/developerworks/library/ar-archtemp/index.html>
- [4] K. Laskey *et al.*, "Reference architecture foundation for service oriented architecture version 1.0," Oasis, Committee Draft 02, October 2009. [Online]. Available: <http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/soa-ra-cd-02.pdf>
- [5] A. Arsanjani *et al.*, "Soa reference architecture," The Open Group, <http://www.theopengroup.org/projects/soa-ref-arch/>, Tech. Rep., September 2009.
- [6] J. de Bruijn *et al.*, "Web service modeling ontology (wsmo)," October 2006. [Online]. Available: <http://www.wsmo.org/TR/d2/v1.3/20061021/>
- [7] D. Martin *et al.*, "Owl-s: Semantic markup for web services," November 2004.
- [8] P. Hill, "Tangibles, intangibles and services: a new taxonomy for the classification of output," *Canadian Journal of Economics*, vol. 32, no. 2, pp. 426–446, April 1999.
- [9] A. Gangemi, P. Mika, and D. Oberle, "An ontology of services and service descriptions," Laboratory for Applied Ontology (ISTC-CNR), Tech. Rep., November 2003.
- [10] R. Ferrario and N. Guarino, "Towards an ontological foundations for services science," in *Proceedings of Future Internet Symposium 2008*, D. Fensel and P. Traverso, Eds. Springer Verlag, 2008.
- [11] L. O. Bonino da Silva Santos *et al.*, "Towards a goal-based service framework for dynamic service discovery and composition," in *Proceedings of the 2009 Sixth International Conference on Information Technology: New Generations*, 2009.
- [12] G. Guizzardi, "Ontological foundations for structural conceptual models," Ph.D. dissertation, University of Twente, 2005.
- [13] L. O. Bonino da Silva Santos *et al.*, "Gso: Designing a well-founded service ontology to support dynamic service discovery and composition," in *2nd International Workshop on Dynamic and Declarative Business Process (DDBP 2009)*, September 2009.
- [14] G. Guizzardi, R. Falbo, and R. S. S. Guizzardi, "Grounding software domain ontologies in the Unified Foundational Ontology (UFO): The case of the ODE software process ontology," in *1th Iberoamerican Workshop on Requirements Engineering and Software Environments (IDEAS'2008)*, Recife, Brazil, 2008.